PROJECT REPORT (Fall 2021-22)

ELL823: Sparse Representation and Compressive Sensing

Done by

Nimesh Gajendraprasad Agrawal (2021EEY7515)

Under the guidance of

Prof. Surendra Prasad,

Department of Electrical Engineering,

Indian Institute of Technology Delhi (IIT Delhi).

November 27, 2021

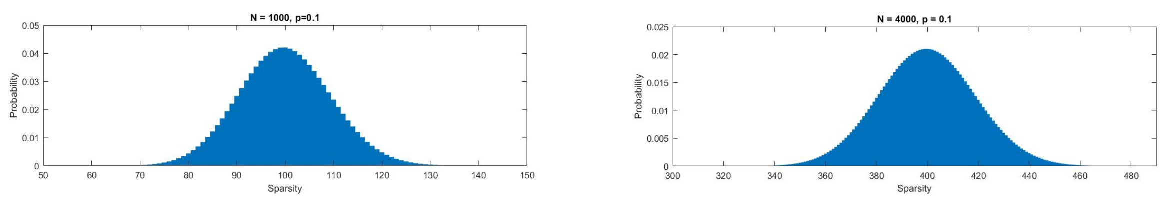
1 Abstract

An experiment for the Recovery of Exact Sparse Signal using the Orthogonal Matching Pursuit(OMP) Algorithm is presented. Initially a random K sparse input signal with N values is being generated. M measurements of this input signal are made by creating a random matrix. Using recovery algorithm based on OMP, nature of recovery error as a function of M is studied for the given values of N and K. Results for few cases are presented to illustrate the effectiveness of the technique.

2 Input Signal Generation

Consider the K sparse input signal $x \epsilon \mathbb{R}^N$, whose support S is determined using Bernoulli Sequence from the binary field $\{1, 0\}$, such that support s(n) = 1 with probability p, where s(n) = 1 indicates that n^{th} value is non-zero. Now the non-zero values are chosen randomly from Standard Normal Distribution i.e $\mathbb{N}(0, 1)$.

The typical range of the values of K are being observed from the binomial distribution curve considering N trials with probability of success p for sparsity level being K. For example, for the case of N = 1000 and 4000, with p = 0.1, the distribution is shown in Figure 1



(a) (b)

Figure 1: (a) N = 1000 (b) N = 4000

It can be observed that for N = 1000 with p = 0.1, K typically lies between 70 to 130 with average sparsity being 100 which is true as the expected value for binomial random variable is (N*p). Similarly for N = 4000 and p = 0.1, K lies between 340 to 460 with average being 400. The typical range and average values of K for parameters N and p are tabulated in Figure 2.

p	N	Range	Average value
0.1	1000	70 to 130	100
	4000	340 to 460	400
0.2	1000	160 to 240	200
	4000	720 to 880	800
0.5	1000	450 to 550	500
	4000	1900 to 2100	2000

Figure 2: Sparsity Levels

3 Sensing Matrix Generation

The generated input signals are being measured by a random sensing matrix $\Phi \epsilon \mathbb{R}^{M \times N}$. The entries of ϕ are random values drawn from Gaussian distribution i.e $\mathbb{N}(0, 1)$. The measured values are than calculated using $y = \Phi x$, where $y \,\epsilon\, \mathbb{R}^M$ is the measurement vector. Since the values of M and N are large, the sensing matrix can be depicted as matrix of dots(pixels) whose grey levels are proportional to the corresponding elements of ϕ. For example, for the case of M = 200 and N = 1000, sensing matrix can be visualized as shown in Figure 3 using the MATLAB command $imagesc(\Phi)$ including $colorbar$.

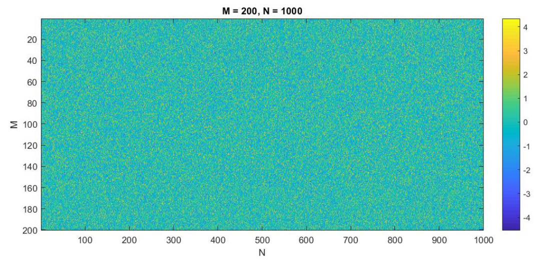


Figure 3: Sensing Matrix

4 OMP for Sparse Recovery

Mainly there are two objectives for sparse recovery algorithm:

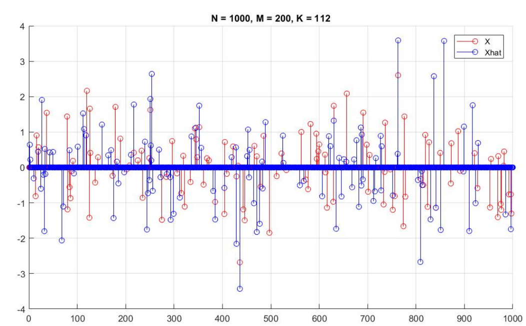
- To identify the locations at which $\hat{x}$ has non-zero values which also corresponds to the sparse support of x.
- Estimation of the values of non-zero values in $\hat{x}$.

x (input signal), ϕ (sensing matrix) and y (measurement vector) are known to us. We want to estimate approximation of x which is denoted as $\hat{x}$. For the given estimate $\hat{x}$, we compute the residual $r = y - \Phi\hat{x}$ during the recovery process. We want to reduce this measurement error norm as much as possible.The recovery process tends to minimize the 2-norm of the recovery error $x - \hat{x}$.

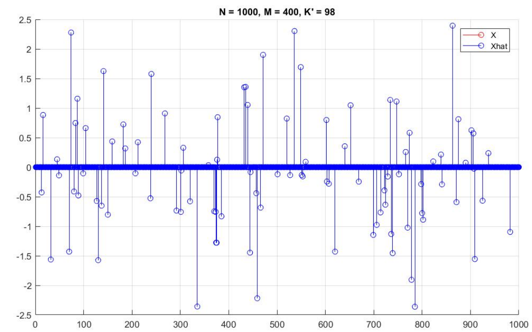
The support after k^{th} iteration is denoted by Λ^k, recovered signal is $\hat{x}^k$ and residual is r^k. Note that r^k is orthogonal to Φ_{Λ^k}. This is true due to $\hat{x}^k$ being the least square solution with $supp(\hat{x}) = \Lambda^k$. This ensures that in each iteration a new column is chosen in $(k+1)^{th}$ iteration.

After implementing the code based on OMP in MATLAB, the recovered signal is compared with the original input signal for different value of number of measurements M for the given values of K and N as parameters. The 2-norm of residual error going below the chosen value of the error threshold is kept as terminating condition for the loop in the code. Consider the case of N = 1000 with p = 0.1. For M = 200, 400 and 600 the recovered signal($\hat{x}$) is plotted with input signal(x) as shown in Figure 4.

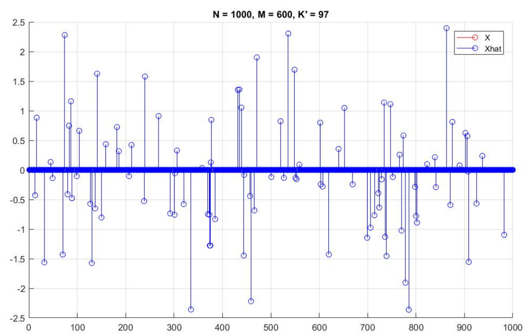
The sparsity level of random input signal generated is K = 97. Here the error threshold is set to 0.01. It is evident that unique recovery of signal x from measurements y is possible iff $spark(\Phi) > 2K$. Now as $spark(\Phi) \,\epsilon\, [2, M+1]$ we want $M \geq 2K$. This result does not guarantee that recovery is possible but it guarantees that unique solution is possible. Hence M = 2K is the minimum requirement. It can be clearly observed from the results that for K = 97, $M \geq 194$ for unique recovery to be possible. For M = 200, the recovery error is quite large. While for both M = 400 and 600, the recovery is exact with error being 0. Also note that the sparsity level of recovered signal denoted by K' is very far from the original signal for the case of M = 200 as algorithm chooses more columns of ϕ while updating support to reach the desired threshold, while it is close for M = 400 and 600.



(a)



(b)



(c)

M	Recovery error (2-norm)	K'
200	11.0319	176
400	0	98
600	0	97

(d)

Figure 4: Comparison of x and $\hat{x}$ for N = 1000 and p = 0.1

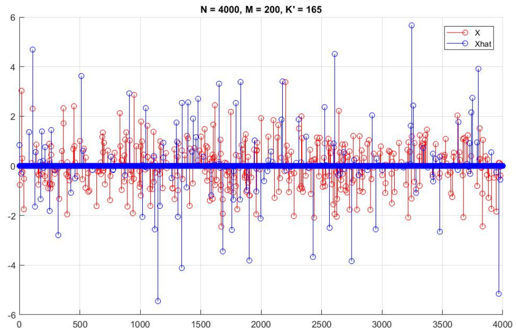
Similarly consider the case of N = 4000 with p = 0.1 and error threshold being 0.01. The sparsity level of random input signal generated is K = 388. Hence $M \geq 776$ for unique recovery to be possible. Hence for the given set of measurements, exact recovery is not achieved as observed from Figure 6. The recovery error in each case is tabulated in the figure. It can be observed that error is reducing with the increase in number of measurements and can only reach 0 for some value of $M \geq 776$.

Now as we increase the probability p to 0.2 and 0.5, it can be observed from the Figure 2 that average values of K increases. Thus, it is not possible to recover the signal with small number of measurements. The results for both this cases are tabulated in Figure 5.

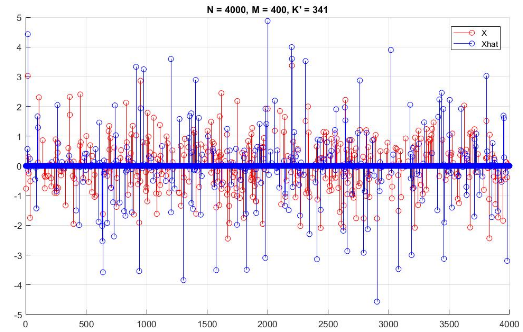
		p = 0.2			p = 0.5		
N	M	K	Recovery error (2-norm)	K'	K	Recovery error (2-norm)	K'
1000	200	190	18.9124	178	504	36.6227	185
	400		8.7253	372		30.1289	379
	600		0	190		21.3973	579
4000	200	745	39.5466	167	1987	70.8003	173
	400		39.3232	346		68.2735	350
	600		37.2709	533		66.1915	537

Figure 5: Results for p = 0.2 and 0.5

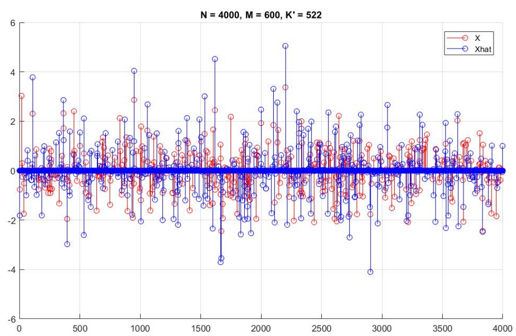
It can be concluded from this results that exact recovery is possible only with case of N = 1000, M = 600 with p = 0.2, while others having large recovery error as the sparsity level is high.



(a)



(b)



(c)

M	Recovery error (2-norm)	K'
200	27.5222	165
400	26.2609	341
600	23.5075	522

(d)

Figure 6: Comparison of x and $\hat{x}$ for N = 4000 and p = 0.1

5 Conclusion

While OMP is provably fast and can also lead to exact recovery, the guarantees associated with OMP for sparse recovery are weaker than those accompanying optimization techniques like l_1 regularization. More precisely, the recovery guarantees are non-uniform,i.e it cannot be shown that a single sensing matrix with $M \geq 2K$ rows can be used to recover every possible K sparse signal with $M \geq 2K$ measurements. But is also possible to obtain such uniform guarantees when it is acceptable to take more measurements. Another issue with OMP is robustness to noise because it is unknown that how solution is disturbed with addition of small amount of noise in measurement. Nevertheless, OMP is an efficient method for sparse recovery, especially when the signal sparsity K is low.